

I hereby certify that this paper is being deposited with the United States Postal Service as Express Mail in an envelope addressed to: Asst. Comm. for Patents, Washington, D.C. 20231, on this date.

June 13, 2000
Date


Express Mail Label No.:
EL409494065US

1 SYSTEM AND METHOD FOR MANAGING
2 MAINTENANCE OF BUILDING FACILITIES

3 The present invention generally relates to a system and a
4 method for managing maintenance of building facilities. More particularly,
5 the invention relates to a system and a method for managing maintenance
6 of building facilities using data transfer between a server computer and a
7 plurality of client computers, each having a unique login identity.

8 The management of maintenance in building facilities
9 typically involves several processes that often have not been effectively
10 defined or integrated with one another in the past. To effectively carry out
11 such management processes, there must be a process for inspecting the
12 building facilities to determine what tasks have been done and what tasks
13 need to be done, and these tasks can change according to various schedules.
14 To have an efficient overall system, other effective processes must be
15 provided, including a work request process, a work order process, a
16 schedule tracking process, and a notification process. In most known
17 conventional systems, most of these processes have been done manually in
18 that they have been at least partially done with the use of considerable
19 human intervention, which is inefficient, costly and time consuming. In
20 addition, massive human intervention in a complex system tends to cause
21 more errors. For example, a customer's inspection orders, work requests,

1 and work orders have often been lost or not processed, which often results
2 in considerable customer dissatisfaction.

3 There are computer-implemented systems that attempt to
4 reduce the use of human intervention at various steps of such complicated
5 building facilities management maintenance. However, these systems do
6 not fully integrate the whole maintenance system. Most of them simply
7 allow customers to communicate with vendors over the web, and do not
8 provide an integrated system that minimizes the need for unnecessary
9 human intervention. Such systems often are unable to offer customization
10 of each building facility that can result in greater precision and organization
11 in the maintenance management system.

12 Accordingly, it is a primary object of the present invention to
13 provide an improved system and method for automatically and efficiently
14 managing maintenance of building facilities with minimal human
15 intervention.

16 Another object of the present invention is to provide such an
17 improved system and method that permits robust customization that can be
18 tailored to each building facility.

19 Still another object of the present invention is to provide such
20 an improved system and method that permits simple and quick
21 communications between the customer and the vendor.

22 Yet another object of the present invention is to provide such
23 an improved system and method that can use mobile computing devices
24 that can configured to display selective data for each assigned job site.

25 A further object of the present invention is to provide such an
26 improved system and method that can operate on a worldwide scale using a
27 large-scale network, such as the Internet.

BRIEF SUMMARY OF THE INVENTION

The present invention is directed to a method and a system for managing maintenance of a building facility using data transfer between one or more server computers and one or more client computers. The system and method allows full integration of the maintenance management process using a computer-implemented system to minimize human intervention and increase system efficiency and accuracy, while reducing operating costs.

More particularly, the system and method are adapted to utilize one or more client computers connected via a large-scale network such as the Internet to a server with a central database. Such client computers can be personal computers, other computers, or mobile computing devices, such as PDA's as well as cell phone devices. Any of these devices will hereinafter be referred to simply as a client. From the central database, job sites are tracked and monitored for fulfillment of inspections, work requests, work orders, or any other scheduled items. In addition, the present invention can automatically send out requests or events responsive to data stored in the central database without human intervention. As a result, the system is adapted to customize each job site and maintain precision and organization with minimized human intervention.

In accordance with an important aspect of the present invention, the system includes one or more server adapted to receive events from a client and forward the events to a clearinghouse via a communication link, one or more client having a unique login identity and adapted to selectively send events to the server via the communication link, and a clearinghouse connected to each server and each client via the communication link for selectively storing data from each server and each client in a database. The clearinghouse is further adapted to selectively authorize predetermined events by each client according to the login

1 identity of each such client, to selectively schedule predetermined events in
2 response to data stored in the database and to monitor the status of all
3 events stored in the database.

4 DESCRIPTION OF THE DRAWINGS

5 FIGURE 1 is an exemplary schematic diagram of a network
6 system in which the present method can be implemented;

7 FIG. 2 is a schematic diagram of the components
8 implemented in the present invention;

9 FIG. 3 is a flow chart illustrating the overall general scheme
10 of the present invention;

11 FIG. 4 is a flow chart illustrating a preferred mobile
12 computing device session;

13 FIG. 5 is a flow chart of the download-tasks event;

14 FIG. 6 is a flow chart of the upload-tasks event;

15 FIG. 7 is a flow chart of the perform-task event;

16 FIGS. 8a through 8d illustrate example displays of the mobile
17 computing device;

18 FIG. 9 is a flow chart illustrating an overall scheme of a
19 session initiated by a user (i.e., a maintenance person) of a client, not using
20 a mobile computing device with preloaded software described in FIGS. 4 to
21 8, for connection with the server;

22 FIG. 10 is a flow chart of the job site-setup event;

23 FIG. 11 is a flow chart the contact-setup event;

24 FIG. 12 is a flow chart the vendor-setup event;

25 FIG. 13 is a flow chart of the inspection-setup event;

26 FIG. 14 is a flow chart of the checklist-item-setup event;

27 FIG. 15 is a flow chart of the performance-rating-method-
28 setup event;

1 FIG. 16 is a flow chart of the performance-rating-type-setup
2 event;
3 FIG. 17 is a flow chart of the special-action-setup event;
4 FIG. 18 is a flow chart of the inspection-templates-setup
5 event;
6 FIG. 19 is a flow chart of the schedule-setup event;
7 FIG. 20 is a flow chart of the inspection-processing event;
8 FIG. 21 is a flow chart of the notification event;
9 FIGS. 22a through 22c comprise a flow chart of the work-
10 request event;
11 FIG. 23 is a flow chart of the work-request-processing event;
12 FIGS. 24a and 24b comprise a flow chart of the work-order
13 event;
14 FIG. 25 is a flow chart of the work-order-processing event;
15 FIG. 26 is a flow chart of the general scheme of the
16 scheduling process; and,
17 FIG. 27 is a flow chart of the general scheme of the
18 monitoring process.

19 DETAILED DESCRIPTION OF THE INVENTION

20 Broadly stated, the present invention is directed to a system
21 and a method for managing operational facilities; the system or method
22 being of the type that utilizes predefined events to carry out managing
23 operations for the facilities. The system includes at least one server adapted
24 to receive events from a client and forward the events to a clearinghouse via
25 a communication link. In addition, the system includes at least one client,
26 but preferably hundreds if not thousands of clients, each of which having a
27 unique login identity, adapted to selectively send events to the server via
28 the communication link. Also included in the system is a clearinghouse
29 connected to each server and each client via the communication link for

the present invention, and these other alternatives or modifications are within the scope of the present invention.

Turning now to FIG. 1, the system in which the present method can be implemented is generally indicated as part of a preferably wide area network 10. A plurality of client computers ("clients") 12 is connected to a plurality of network servers ("server") 14 via the network 10. As an example, the clients 12 can be network servers, which in turn are connected to workstations 16 within an intranet. In addition, the present invention can be implemented using a variety of connections, such as the Internet or wireless communication system. The connection functions primarily to allow the server and the client to communicate and transfer data preferably but not necessarily using real time communication.

However, the Internet is the preferable network connection 8 because it provides the most flexible and universal way of communicating. If the Internet is used as the communication connection between the client 12 and the server 14, Extensible Markup Language (XML) is the preferred language for its implementation. However, the present invention can be implemented practically in any number of ways that may evolve with evolving technology. To further the complexity of the various network types that may be available, issues of bandwidth, reliability and security of the network are important considerations. As a result, an explanation of the current preferred embodiment of the network topology is given as an example and other networks and connections are within the scope of the present invention.

A schematic diagram of the components implemented in the present invention is shown in FIG. 2, and includes a clearinghouse 18 linked to a database 20. The database 20 acts as a central storage location for the data 22 for each building facility. The clearinghouse 18 is, among other things, the management system for the present invention. The clearinghouse is directly linked to the database 20 for inputting and

portable devices with wireless Internet access, such as a Personal Digital Assistant or Pocket PC. The MCD, in this case, responds or sends events preferably by connecting to the web page directly on the MCD. As a result, there is no need to preload the MCD with software or data. The needed data will be displayed through the web page. In this instance, the MCD does not need the process described in FIG. 4. This alternative implementation is within the scope of the present invention.

The flow chart of FIG. 4 illustrates a preferred mobile computing device (“MCD”) session, which is triggered by the device user (block 42). The process begins (block 44) by the user starting a client using software that was previously installed and specifically designed for the MCD 12, which will be referred to as the task list program (“TLP”) (block 46). The MCD 12 next enters a username and a password to sign onto the server 14 (block 48), and checks whether the login was successful (block 50). If the login was unsuccessful, the MCD 12 displays an error message from the server 14 explaining why the login failed (block 52).

If, on the other hand, the login was successful (block 50), the MCD 12 will establish a connection with the server and start communicating with the server for downloading of the task list (block 54). The MCD 12 then downloads the task list in an in-box designed for the login identity of the user from the server (block 56), which initiates a download-tasks event (block 58) shown in FIG. 5 and will be explained in greater detail below. After the download-tasks event has been processed (block 58), the MCD next uploads completed tasks in its out-box for this user (block 60), which initiates the upload-tasks event (block 62) shown in FIG. 6 and will again be explained later in great detail.

Because a successful login gives the MCD a specific user identity matching stored information in the database, the user of the MCD still has a choice to download tasks, upload tasks, perform a task, or exit the TLP (block 64) with or without a specific user identity. If the user wants to

1 download tasks from the server (block 64), the download-tasks event is
 2 initiated (block 66). Similarly, the upload-tasks event is initiated (block 62)
 3 if the user chooses to upload tasks to the server (block 68). If the user
 4 chooses to perform a specific task on the MCD (block 64), a perform-task
 5 event is initiated (block 70) which is shown in FIG. 7 and will be described
 6 in detail. Finally, the user can also exit the process (block 72) by choosing
 7 to exit the TLP (block 64).

8 A flow chart of the download-tasks event 66 is shown in FIG.
 9 5, and is initiated by the MCD (block 74), and the process begins (block 76)
 10 with the MCD checking the connection with the server (block 78). If the
 11 MCD is not connected, the user has to login using a username and
 12 password (block 80). Then, it is checked again if the login was successful
 13 (block 82). If not, the MCD displays an error message from the server
 14 (block 52) and brings the user back to the choices of downloading tasks,
 15 uploading tasks, perform a task, or exit TLP (block 64). Otherwise, once
 16 the connection with the server is established (blocks 78, 82), the next step is
 17 to move a task pointer to the beginning of the new task list (block 84). The
 18 MCD 12 downloads the first task in the task list from the in-box for this
 19 user (block 86) and determines whether the downloaded task list data is
 20 valid (block 88). If the data are valid (block 88), the MCD determines
 21 whether that is the end of the task list (block 90). If so, the process ends
 22 (block 92). Otherwise, it loops back and downloads the next task from the
 23 list (block 94).

24 On the other hand, if data is invalid on the task list (block 88),
 25 the MCD is prompted to determine whether the connection with the server
 26 is still valid (block 96). The MCD will make an entry in the exception log
 27 (block 98) if the connection is no longer valid (block 96), and the process
 28 ends (block 92). Otherwise, when the connection is still valid (block 96),
 29 the MCD increments a retry count (block 100) and determines whether the
 30 incremented retry count has reached its predefined maximum number of

1 retries (block 102). Again, if maximum number of retries has been reached
2 (block 102), an entry in the exception log will be made (block 98) and the
3 process ends (block 92).

4 A flow chart for the upload-tasks event 68 is shown in FIG. 6
5 and is triggered by the TLP (block 106) and starts the process (block 108).
6 It is first determined whether the MCD is connected to the server (block
7 110). If not, the user must enter a username and a password to establish a
8 connection with the server (block 112). However, if the login is
9 unsuccessful (block 114), the process loops back to display an error
10 message from the server to the user (block 52), after which the user is given
11 the option to choose whether to download tasks, upload tasks, perform a
12 task, or exit TLP (block 64).

13 If the login is successful (block 114), similar to the download-
14 tasks event 72, the task pointer moves to the beginning of the completed
15 task list (block 118) to ensure that the first completed task is uploaded. The
16 MCD 12 uploads the first task in the list from the out-box for this user
17 (block 118), and determines whether the upload is successful (block 120).
18 The MCD 12 determines whether this is the end of the task list (block 122)
19 and if the upload was successful (block 120). If so, the process ends (block
20 124). Otherwise, it loops back and downloads the next task from the list
21 (block 126).

22 On the other hand, if the upload proves to be unsuccessful
23 (block 120), the MCD is prompted to verify that the connection with the
24 server is still valid (block 128). The MCD will make an entry in the
25 exception log (block 130) if the connection is not valid (block 130), and the
26 process ends (block 124). Otherwise, when the connection is still valid
27 (block 128), the MCD increments a retry count (block 132) and determines
28 whether the incremented retry count has reached its predefined maximum
29 number of retries (block 134). Again, if maximum number of retries has

1 been reached (block 134), an entry in the exception log will be made (block
2 130) and the process ends (block 124).

3 Referring to FIG. 7, which illustrates a flow chart of the
4 perform-task event 70, the user first selects a task from the task list stored
5 in the MCD (block 138) and elects to perform the selected task (block 140).
6 Then, the MCD opens an associated task execution program ("TEP"),
7 which runs the MCD from this point (block 142). The TEP is generally a
8 program displaying a specific template or form that is designated to the
9 selected task with its customization. For example, if the selected task is for
10 a specific job site having predefined custom checklist items for an
11 inspection, then the TEP displays the form with the predefined checklist
12 items for the user to complete the inspection as requested by the job site.
13 This allows for customization and provides simpler ways to accomplish a
14 specific task, because the MCD can display the correct forms that match the
15 selected task to the user. In this example, the user is shown a series of
16 screens necessary to accomplish the selected task (block 144). However, at
17 any given point during this process, the user can choose to complete the
18 task or stop and return to the TLP (block 146).

19 The TEP displays the first checklist item of the selected task
20 for a response from the user (block 148). The user can then choose to
21 respond, stop or skip this particular checklist item (block 150). If the user
22 chooses to stop the TEP (block 152), the TEP ends and passes control back
23 to the TLP (block 154). On the other hand, if the user does not choose to
24 stop the TEP (block 152), the user must choose to either skip or respond to
25 the checklist item. The TEP stores the response for this checklist item if
26 the user responds to the checklist item (block 155) and proceeds to the next
27 checklist item once that is done (block 156). However, the next checklist
28 item is still displayed (block 156) even if the user chooses to skip this
29 checklist item. It is then determined whether the selected task is completed
30 (block 158). In other words, the routine determines whether the user has

1 processes, such as a jobsite-setup event, be done in the web browser
2 environment.

3 As an example, the process is triggered by the client opening
4 the home page provided for the implementation of the present invention
5 (block 160). Although the preferred connection is through a web page
6 setting, it is not necessary. For example, the present invention can be
7 implemented using other connections, such as a private network. These
8 alternative connections are within the scope of the present invention.
9 However, a XML web page environment is preferred because it can
10 presently provide the most flexible and simplest environment for the
11 implementation of the present invention. Regardless, the preferred
12 environment can change with technology, and other possible alternatives
13 are also within the scope of the present invention.

The session begins (block 162) with the user first providing a username and a password in order to log into the server and becomes an authorized client (block 164). The server next determines whether this is a valid user (block 166). If not, the server makes an entry in the security log (block 168) and the process ends (block 170). The user can then choose an option from the menu (block 172). The options include initiating a job site-setup event 28, an inspection-templates-setup event 174, work request event 26, work order event 38, schedules-setup event 32, and contacts-setup event 176. Once the selected option has been processed, the session checks if the authorized client wants to continue with choosing the options in the menu (block 178). If so, it loops back to the option menu (block 172). Otherwise, the session ends (block 170).

A more detailed description of the jobsite-setup event 28 previously described in FIGS. 3 and 9 is shown in the flow chart of FIG. 10. This event is generally initiated by the user of the client 12 (block 180). The process begins (block 182) by giving the user of the client a choice of adding new data, editing existing data, or exiting from the event (block

1 184). If exiting the job site-setup event is selected, the process ends at that
 2 point (block 186). The server displays the existing data to the client for
 3 revision for the choice of editing existing data, and the process continues on
 4 to the next step when the option of adding new data is selected (block 188).
 5 The user generally first adds or revises the job attributes, the name and
 6 address of the building facility or other information that helps identify the
 7 building (block 190). The user next adds or revises the identification of the
 8 various parts, sections, or areas of the job site (block 192).

9 Next, the user sets up the contacts for the job site (block 194),
 10 which will initiate the contact-setup event shown in greater detail in FIG.
 11 11. Then, the user sets up the vendors for the job site (block 196) initiating
 12 the vendor-setup, which is shown in FIG. 12. After that is done, the user
 13 has to set up the inspections initiating the inspection-setup event (block
 14 200) and special actions initiating the special-actions-setup event (block
 15 204). For each inspection that is setup, the user defines a schedule for the
 16 inspection (block 208), which is followed by defining defaults for any
 17 information needed for the job site (block 210). The server 14 saves all the
 18 information onto the database (block 212). It is next determined whether
 19 the user wants to continue setting up another job site (block 214). If so, the
 20 process loops back to the option menu (block 184). Otherwise, the process
 21 ends (block 186).

22 The contact-setup event 194 is shown in more detail in the
 23 flow chart of FIG. 11, and is generally initiated by the user of the client
 24 (block 216). However, as shown in FIG. 10, it can also follow from
 25 another event. The process begins (block 218) initially by giving the user
 26 an option menu for adding new data, editing existing data, or exiting the
 27 contact-setup event (block 220). If exiting the contact-setup event is
 28 selected, the process will end (block 222). Otherwise, the server displays
 29 the existing data to the client for revision (block 224) when editing existing
 30 data is selected, and the process continues on to the next step when adding

18

With regard to the inspection-setup event 200 (FIG. 10), it is shown in more detail in FIG. 13, and is generally initiated by the client (block 258). Similarly, the process starts (block 260) with an option menu for adding new data, editing existing data, and to exit the inspection-setup event (block 262). If exiting the inspection-setup event is selected, the process ends (block 264). Otherwise, the server displays the existing data to the client for revision if editing existing data is selected (block 266), and the process continues on to the next step when adding new data is selected. The user generally first adds or revises the inspection attributes, description, or other useful information about the inspection (block 268). For each area of the job site, the user defines the inspection steps using items from an existing checklist for this job site or from a default inspection template stored on the database (block 270). The user next revises the checklist items as needed (block 272), and the checklist-item-setup event 274 is initiated (block 276). The checklist-item-setup event 274 is shown in FIG. 14, and will be described below in greater detail. The server saves the revised inspection data including inspection records and schedules onto the database (block 278), and determines whether the user wants to continue with the inspection-setup event (block 280). If so, the process goes back to the option menu (block 262). Otherwise, the process ends (block 264).

The checklist-item-setup event 274 previously mentioned in FIG. 12 is shown in more detail in the flow chart of FIG. 14. Although this event follows from the inspection-setup event, the event can be initiated by the client at any point in the whole system (block 282). The process starts (block 284) similarly with an option menu for adding new data, editing existing data, or exiting the contact-setup event (block 286). If the user chooses to exit the checklist-item-setup event, the process ends (block 288).

1 initiated (block 316). Again, the server saves the revised information onto
2 the database (block 320), which is followed by a step determining whether
3 the user wants to continue with the current event (block 322). If so, the
4 process goes back to the option menu (block 308). Otherwise, the process
5 simply ends (block 310).

More specifically, with regard to selecting the performance rating type block 316 and referring to FIG. 16, it is triggered by the performance-rating-method-setup event (block 324). The process starts with an option menu of three performance rating types (block 328), specifically yes/no, multiple options, and numeric scoring. For the yes/no type, there will be only two valid responses (e.g., done/not done) in which the user will specify the valid responses (block 330). However, the two valid responses are mutually exclusive, meaning the logic should prevent it from choosing both at the same time. Next, for the multiple options type, any number of options (e.g., good, fair, or poor) are possible (block 332), and the user has to specify each available option in this setup. But only one option is allowed as a valid response. Similarly, the logic allows only one option to be chosen. With the numeric type, the user specifies a range of numeric values having a minimum and a maximum along with a step interval (e.g., 1 to 10 with an increment of 0.5) (block 334). The process ends (block 336) when the user finishes selecting and defining the performance rating type.

The routine for setting up a special-action event 204 is shown in FIG. 17, which is triggered by the job site-setup event 28 in FIG. 10 (block 338). The process begins (block 340) with the user defining a special action for an individual checklist item, a group of checklist items, or a job area (block 342). The user can also define special actions for each of the three performance rating types discussed in FIG. 16 (block 344). Alternatively, if the special actions are based on a group, the total score for that group can be used as a threshold comparison for the special actions

1 (block 344). Next, the user can set up one or more actions for each special
2 action, such as notifying one or more contacts, creating a work request or
3 work order and notifying the contacts (block 346). And each special action
4 can also include a response time (block 346). The user can also setup the
5 special action to be triggered by the response from the three performance
6 rating type. For the yes/no type of action, it can be triggered by either of
7 the events (block 348). Similarly, within the multiple options type, the
8 action can be triggered by the valid response (block 350). Finally, for the
9 numeric type, the user can use a range method and define the range that
10 triggers the special actions (block 352). With that, the process ends (block
11 354).

12 To setup the inspection-templates-setup event 174 as
13 described in FIG. 9, the steps of the flow chart shown in FIG. 18 are carried
14 out. The event can also be triggered by the user of the client 12 who
15 chooses this option on the web page (block 356). The process starts (block
16 358) with an option menu of adding new data, editing existing data, or
17 exiting from the event (block 360). If exiting the job site-setup event is
18 selected, the process ends (block 362). Otherwise, either the server
19 displays the existing data to the client for revision if the edit option is
20 chosen (block 364), or the process continues on to the next step if the add
21 option is chosen. The user then adds or revises the template attributes and
22 description, or other information that may be useful (block 366). The user
23 next defines the inspection template as the inspection steps using items
24 from the checklist previously setup in FIG. 14 (block 368). The checklist
25 items can either be from an existing checklist or a default inspection
26 template stored on the database. These are the checklist items that are
27 displayed for a selected job area (block 192) in the previous discussion of
28 the perform-task event in FIG. 10. The user at this time can edit the
29 checklist items (block 370). The server then saves the revised inspection
30 template data onto the database (block 372), and determines whether the

To setup the schedules 32 (FIG. 3), and referring to the flow chart of FIG. 19, this event is generally triggered by an external process, such as the one described previously in FIG. 9 (block 376). However, the event can also be triggered by the user of the client 12 choosing this option on the web page. The process starts (block 378) with an option menu of adding new data, editing existing data, or exiting from the event (block 380). This event allows the user to setup different schedules and associate them with a selected event, such as an inspection event. The process ends (block 382) if the user chooses the exit option. However, if the user chooses the edit option, the server displays the existing schedule data stored in the database to the client for revision (block 384). Alternatively, the process continues on to the next step if the add option is selected, which is adding and revising the schedule attributes and description or other useful information (block 386). The user next defines the schedule parameters, such as the frequency of the event and whether it is rule based or fixed dates based (block 388). The server saves the revised schedule data onto the database (block 390), and again determines whether the user wants to continue with the schedule-setup event (block 392). If so, the process goes back to the option menu (block 380). Otherwise, the process ends (block 382).

The detailed steps of carrying out an inspection-processing event 40 from FIG. 3 is shown in the flow chart of FIG. 20. This event is generally triggered by the client sending inspection data to the server (block 394) either on the web page or the MCD. More specifically, the client generally sends inspection data to the server, for example, once an inspection is completed by the user on the MCD. The process begins (block 396) by validating the sent data (block 398). If it is found that the

1 data is invalid (block 400), the server will make an entry in the exception
 2 log (block 402) and the process ends (block 404). On the other hand, if the
 3 data is valid (block 400), the server will save the inspection data on the
 4 database (block 406). The inspection data is then compared with the
 5 allowed pre-defined tolerances from the performance-rating data (block
 6 408) to determine whether the data is within the tolerances (block 410). If
 7 so, the server initiates a notification event 34 (block 412) that allows the
 8 server to send a message informing a contact person of the job site of the
 9 inspection being within preset tolerances.

10 If the routine does not end (block 404), meaning that the data
 11 is not within the predefined tolerances (block 410), it is next determined
 12 whether any special action is required (block 414). Whether any special
 13 action is required depends on the data that were previously defined in all
 14 the setup events. The clearinghouse evaluates the data and makes decisions
 15 on certain actions based on all the data stored in the database, and sends
 16 them to the server for performance when necessary. If no special action is
 17 required (block 414), the server again initiates a notification event 34 to the
 18 clearinghouse for sending a message to the contact of the status of this
 19 inspection (block 416). However, if special actions are needed (block 414),
 20 the server will either initiate a work-request event 26 (block 418) or a work-
 21 order event 38 (block 420), depending on the instructions from the data
 22 stored in database. When either events are initiated, a message will be sent
 23 to the contact person initiating the notification event 34 (block 422 and
 24 424) to bring the process to an end (block 404).

25 With regard to the notification event described in connection
 26 with FIG. 20, and referring to the flow chart of FIG. 21, the notification
 27 event can be initiated at any time during operation of the system. Basically,
 28 it is initiated whenever a message is being sent to a specific recipient, the
 29 identity of which is dependent on information stored in the database. The
 30 process starts (block 426) with the server retrieving the recipient

1 Furthermore, after the notification to the user (block 462) or
2 the required job is not in the list (block 460), the server inquires whether
3 the user wants to specify a location (block 464) as shown in FIG. 22(b). If
4 the user does not want to specify a location (block 464), the server next
5 inquires whether the user wants to send a message (block 466). If not, the
6 process ends at this point (block 454). Otherwise, the user can compose a
7 message to the default contact for a work ticket without any job
8 information, or an alternative contact of the user's choice (block 468). As a
9 result, the notification event is initiated because a message is being sent
10 (block 470), and the process comes to an end (block 454).

11 However, if the user wants to specify a location (block 464),
12 the user can enter the location information and description for the work to
13 be performed with the price, payment, terms, approval notice, and due date
14 and time (block 472). At that point, the work request without a job
15 association will be saved on the database, and the default contact will be
16 notified of the work request (block 474). The notification event 34 is
17 initiated by the notification to the default contact (block 476). The last step
18 is to determine whether the user wants to notify another contact (block
19 478). The user picks another contact from a list from the server (block
20 480), and the notification event is initiated again (block 482). Otherwise,
21 the process ends (block 454).

Returning to the beginning of the process in FIG. 22(a), if the required job is in the list (block 460) and the user selects a job from the list (block 484) or an existing work request is displayed to the user (block 487), the user can enter or edit the description of the work to be preformed with the due date and time (block 486). The server determines whether the client wants to contact the default authorized client for this work request (block 488). If not, the client picks another authorized client from the contact list for the job site (block 490). The server then saves the work request with the selected authorized client onto the database (block 492).

1 Next, the server sends the details of the work request to the authorized
2 client (block 494), initiating another notification event (block 496). The
3 process ends at this point (block 454).

4 If the work-request event originates from the MCD, which is
5 different from initiating the event on the web page, the flow chart of FIG.
6 22(c) applies. When the work-request event is triggered by the MCD
7 (block 498), the process starts (block 500) with two options of adding new
8 data or editing existing data, and is generally done during an inspection.
9 For adding new data, the user creates and edits one or more work requests
10 for the job site for which the inspection is in progress (block 502). The
11 MCD will scan the inspection data to establish whether the user has
12 permission to create such a work request (block 504). If the user does not
13 have permission (block 506), the MCD notifies the user (block 508) and
14 asks whether the user wants to send a message (block 510). If so, the user
15 composes the message to the primary contact of the job site (block 512),
16 which the server will send to the contact using the notification event 34
17 (block 514). As shown in FIG. 22(a), the process then ends (block 454).

18 Assuming the server displayed the existing data to the client
19 for revision (block 516) or the client has permission to create a work
20 request (block 506), the user will enter and edit the work request as needed,
21 which might include the description, price, payment terms, approval notice,
22 and due date and time (block 518). This information is then saved to the
23 database by the server after the client has finished revising (block 520), and
24 the revised data is sent to the designated authorized client (block 494) using
25 the notification event 34 (block 496), bringing the process to an end (block
26 454). The use of the authorized client must now process the work request,
27 which is shown in FIG. 23.

28 Turning now to FIG. 23, a flow chart of the work-request-
29 processing event for the designated authorized client of a work request is
30 shown and generally indicated as 522. The event begins (block 524) with

If the user of the authorized client accepts the work request, an approval code must be entered by the user, and then validate the approval by entering either a password or pin number (block 548). The server will then save the information onto the database and create a work order for the approved work request (block 550), which will initiate the work-order event 38 (block 552). If, on the other hand, the work request is rejected, the user preferably requests an explanation and must also validate the request with password or pin number (block 554). The server next determines whether the user of the client asks for a revised work request (block 556). If not, the process ends (block 546). If the user does ask for a revised work request (block 556), the server updates the database with the revised work request (block 558). As a result, a notification message will be sent to the contact, initiating the notification event (block 560). The process then ends (block 546).

With regard to the work-order event, and referring to the flow chart of FIGS. 24(a) and 24(b), this event is triggered by the user choosing

1 this option or by other events (block 562). The process begins (block 564)
2 with an option menu from which the user can select to add new data, edit
3 existing data, or exit the current event (block 566). If the user exits the
4 current event, the process ends (block 568). If the user elects to add new
5 data, the server creates a list of job sites authorized to the current user to
6 add new work orders (block 570). It is next determined whether the list is
7 empty (block 572). If the list is not empty (block 572), it is determined
8 whether the required job is in the list (block 574). If the list is empty (block
9 572), the user is notified of the list being empty (block 576). After the user
10 has been notified (block 576) or the required job is not in the list (block
11 574), the server asks the user if a work request should be created (block
12 578). If the user elects not to create a work request (block 580), the routine
13 ends (block 568). Otherwise, a work request is created initiating the work-
14 request event (block 582) if the user decides to create one (block 580).

15 Returning back to the beginning of the process in FIG. 24(a),
16 if the required job is in the list (block 574) and the user selects a job from
17 the list (block 584) or an existing work request is displayed to the user
18 (block 586), the user can enter or edit the description of the work to be
19 preformed with the due date and time of the work order (block 588). The
20 server inquires whether the client wants to contact the default recipient of
21 the work order (block 590). If not, the client picks another recipient from
22 the contact list for the job site (block 592). The server then saves the work
23 order with the selected recipient in the database (block 594). Next, the
24 server sends the details of the work order to the recipient (block 596),
25 initiating a notification event (block 598). The process ends at this point
26 for the work order event (block 568).

27 To process a work order, and referring to the flow chart of
28 FIG. 25, the work-order-processing event begins (block 602) and is
29 triggered by either the recipient using an authorized client to access the web
30 page (block 604) or data received (block 606). If this is a user initiated

1 session, the server determines whether there is a record identifier available
2 for the work order (block 608). If the authorized client did not supply the
3 server with a record identifier (block 608), the server prepares a list of all
4 open work orders for the authorized client for selection (block 610). The
5 recipient selects a work order from the list (block 612), and the server
6 displays the selected work order to the recipient (block 614). Alternatively,
7 when the data is sent from a recipient using a MCD to the server, for
8 example, the server validates the received data (block 616). If the data is
9 not valid (block 618), the server makes an entry in the exception log (block
10 620) and the process ends (block 622). Assuming that the data is valid
11 (block 618) or that the user updated the existing work order, the server
12 updates this information onto the database (block 624). At this point, a
13 notification is sent to the contact person of the job site using the notification
14 event (block 626), and the process ends (block 622).

15 In accordance with another important aspect of the present
16 invention and referring to FIG. 26, the clearinghouse keeps a scheduling
17 process to respond to the scheduled items from a client or an event. The
18 clearinghouse preferably runs this process continuously. It begins (block
19 628) with a timer (block 630) that triggers the process (block 632). The
20 clearinghouse determines whether the process should end according to the
21 timer (block 634). If so, the process ends (block 636). However, if it is
22 determined that the process should continue (block 634), the clearinghouse
23 reads a scheduled event from a schedule list (block 638) and determines
24 whether this is the end of the list (block 640). If it is the end of the list
25 (block 640), the clearinghouse waits until it is again triggered by the timer
26 (block 632). On the other hand, if it is not the end of the list, the
27 clearinghouse responds to the scheduled event (block 642). How the
28 clearinghouse will respond to the event depends upon the type of event. In
29 addition to responding to the event in the predefined process proposed in
30 each event, the clearinghouse further sends a notification to the default

1 each building facility while maintaining precision and organization in the
2 management system.

3 While various embodiments of the present invention have
4 been shown and described, it should be understood that other modifications,
5 substitutions and alternatives are apparent to one of ordinary skill in the art.
6 Such modifications, substitutions and alternatives can be made without
7 departing from the spirit and scope of the invention, which should be
8 determined from the appended claims.

9 Various features of the invention are set forth in the appended
10 claims.